CLAIMS

What is claimed is:

1. Apparatus for protocol conversion, comprising:

a device emulator coupled to a first device having a first protocol;

digital storage coupled to the device emulator for temporary storage of information from the first protocol;

at least one manager (i) coordinating the transfer of the information of the first protocol between the device emulator and the digital storage and (ii) coordinating transfer of the information between the digital storage and a second protocol.

2. The apparatus as claimed in Claim 1, wherein the device emulator is a tape drive emulator.

3. The apparatus as claimed in Claim 1, wherein the digital storage includes at least one of the following storage devices: magnetic disk, optical disk, and digital memory components.

4. The apparatus as claimed in Claim 1, wherein the manager manages input/output data between a mainframe computer and a commercial queuing system.

5. The apparatus as claimed in Claim 4, further including a group driver (i) supporting at least one pseudo-tape driver and (ii) interfacing with the digital storage.

6. The apparatus as claimed in Claim 1, further including a second device emulator coupled to the digital storage, wherein said at least one manager coordinates transfer of information between the two device emulators.

7. The apparatus as claimed in Claim 1, used in a wide area network to share data among multiple device emulators and at least two protocols.

8. The apparatus as claimed in Claim 1, used to transfer data between or among multiple mainframe computers.

9. The apparatus as claimed in Claim 1, wherein the information is arranged in a queue in the digital storage.

10. A method for protocol conversion, comprising:

emulating a peripheral device to receive information from a first computer having a first protocol;

temporarily storing the information;

coordinating the transfer of the temporarily stored information having a first protocol to a second computer having a second protocol in a manner causing the information to take on characteristics of the second protocol.

11. The method as claimed in Claim 10, wherein said emulating includes emulating a tape drive.

12. The method as claimed in Claim 10, wherein storing the information includes writing the information to at least one of the following storage devices: magnetic disk, optical disk, and digital memory components.

13. The method as claimed in Claim 10, wherein said coordinating the transfer of the temporarily stored information includes managing input/output data between a mainframe computer and a commercial queuing system.

14. The method as claimed in Claim 13, further including directing the information from the first computer to a digital storage area used to temporarily store the information.

15. The method as claimed in Claim 10, further including emulating a second peripheral device and coordinating transfer of information between the first computer and the second computer separated by temporarily storing the information.

16. The method as claimed in Claim 10, used in a wide area network to share data among multiple computers using multiple protocols.

17. The method as claimed in Claim 10, used to transfer data between or among multiple mainframe computers.

18. The method as claimed in Claim 10, wherein the temporarily stored information is arranged in a queue.

19. In an apparatus for protocol conversion, a manager having distributed components, comprising:

at least one I/O manager having intelligence to support states of (i) emulation devices transceiving messages using a first protocol and (ii) an interface transceiving messages using a second protocol;

at least one emulation device providing low-level control reaction to an external device adhering to the first protocol; and

at least one group driver to provide an interface between the I/O manager and said at least one emulation device.

20. The manager as claimed in Claim19, wherein said at least one group driver buffers data to allow for direct memory access (DMA) transfer.

21. The manager as claimed in Claim 19, wherein said at least one emulation device emulates at least one tape drive.

22. The manager as claimed in Claim 19, wherein the I/O application includes multiple input/output managers.

23. The manager as claimed in Claim 19, wherein the manager includes a sufficient number of emulation devices, group drivers, and I/O managers to maximize parallel processing performance of protocol conversion.

24. A method for protocol conversion, comprising:

using an I/O manager, transceiving messages with at least one first external device using a first protocol;

using the I/O manager, transceiving the same messages with at least one second external device using a second protocol;

emulating low-level control reactions to support the transceiving of the messages with the first external device in a manner that disassociates the I/O manager from the low-level control reactions; and

channeling data flow between the I/O manager and said at least one first external device in a manner that minimizes interfacing by the I/O manager with said at least one first external device.

25. The method as claimed in Claim 24, further including buffering data to allow for direct memory access (DMA) transfers.

26. The method as claimed in Claim 24, wherein said emulating low-level control reactions is performed in a manner similar to that of a tape drive.

27. The method as claimed in Claim 24, further including channeling multiple data flows simultaneously by employing multiple I/O managers.

28. The method as claimed in Claim 24, further including a plurality of transceiving, emulating and channeling steps in a parallel manner to maximize parallel processing performance of the protocol conversion.

29. Apparatus for mainframe-to-mainframe connectivity, comprising:

a first device emulator in communication with a first mainframe and acting as a standard sequential storage device;

a second device emulator in communication with a second mainframe and also acting as a standard sequential storage device;

digital storage coupled to the first and second device emulators to store information temporarily for the first and second device emulators; and

at least one manager (i) coordinating a first transfer of information between the first device emulator and the digital storage and (ii) coordinating a second transfer of information from the digital storage to the second device emulator, the first and second mainframes having access to the information via respective device emulators.

30. The apparatus as claimed in Claim 29, wherein the information stored in the digital storage is arranged in a queue.

31. The apparatus as claimed in Claim 30, wherein the length of the queue is short to approach real-time protocol conversion.

32. The apparatus as claimed in Claim 30, wherein said manager dynamically adjusts the length of the queue.

33. The apparatus as claimed in Claim 29, further including a queue manager to support a case in which the first and second mainframes are not synchronized when transferring information via the apparatus.

34. The apparatus as claimed in Claim 29, wherein the second device emulator communicates with a second device emulator of a remote apparatus to transfer the information over a data network to provide remote connectivity between the first and second mainframes.

35. A method for providing mainframe-to-mainframe connectivity, comprising:

assigning a first digital memory region external from a first mainframe to store messages in a sequential order for the first mainframe;

assigning a second digital memory region external from a second mainframe to store messages in a sequential order for the second mainframe;

emulating a device capable of communicating with the first and second mainframes to respond to requests from at least one of the mainframes; and

in response to a request from at least one of the mainframes, establishing a link between the first and second digital memory regions to provide effective mainframe-to-mainframe connectivity between the first and second mainframes.

36. The method as claimed in Claim 35, further including storing messages from the mainframes in the digital memory region in a queue arrangement.

37. The method as claimed in Claim 36, wherein the length of the queue is short to approach real-time protocol conversion.

38. The method as claimed in Claim 36, further including dynamically adjusting the length of the queue.

39. The method as claimed in Claim 35, further including managing the queue to support a case in which the first and second mainframes are not synchronized when transferring information between the first and second mainframes.

40. The method as claimed in Claim 35, wherein emulating a device includes communicating with a remote process also emulating a device to transfer the information over a data network to provide remote connectivity between the first and second mainframes.

41. In a data storage system, a method for managing messages, comprising:

receiving information that is normally contained in a standard tape label;

based on the information, applying the information to a non-tape memory designated for a message queue;

storing messages related to the information in the memory; and

managing the message queue as a function of the standard tape label information.

42. The method as claimed in Claim 41, wherein the information normally contained in a standard tape label includes at least one of the following elements: volume serial number, data set name, expiration date, security attributes, and data characteristics.

43. The method as claimed in Claim 42, further including creating a queue name based on the volume serial number and data set name.

44. The method as claimed in Claim 42, further including deciding how long to maintain the message queue based on the expiration date.

45. The method as claimed in Claim 42, further including securing the message queue based on the security attributes.

46. The method as claimed in Claim 42, further including optimizing the message queue based on the data characteristics.

47. The method as claimed in Claim 42, further including mounting the message queue based on the volume serial number or data set name in response to receiving a request for either.

48. Apparatus for managing messages, comprising:

a receiver to receive information from a computer that is normally contained in a standard tape label; and

a controller that (i) applies the information to a non-tape memory, designated for a message queue, (ii) stores messages related to the information in the memory, (iii) manages the message queue as a function of the standard tape label information.

49. The apparatus as claimed in Claim 48, wherein the information normally contained in a standard tape label includes at least one of the following elements: volume serial number, data set name, expiration date, security attributes, and data characteristics.

50. The apparatus as claimed in Claim 49, wherein the controller creates a queue name based on the volume serial number and data set name.

51. The apparatus as claimed in Claim 49, wherein, based on the expiration date, the controller decides how long to maintain the message queue.

52. The apparatus as claimed in Claim 49, wherein, based on the security attributes, the controller secures the message queue.

53. The apparatus as claimed in Claim 49, wherein, based on the data characteristics, the controller optimizes the message queue.

54. The apparatus as claimed in Claim 49, wherein the controller mounts the message queue based on the volume serial number or data set named in response to receiving a request for either.

55. Apparatus for protocol conversion, comprising:

means for interfacing with a computer having legacy applications;

means for interfacing with an open system network;

means for emulating a sequential storage device in a manner supported by the legacy applications;

means for storing data being transferred between the computer and devices coupled to the open system network, said means for storing data interacting with said means for emulating a sequential storage device; and

means for providing the computer and devices access to the stored data.